A Non-Intrusive Load Monitoring Approach for Very Short Term Power Predictions in Commercial Buildings

Karoline Brucke, Stefan Arens*, Jan-Simon Telle, Thomas Steens, Benedikt Hanke, Karsten von Maydell, Carsten Agert

Abstract

In this study, a new algorithm is developed to extract device profiles in a fully unsupervised manner from three-phases reactive and active aggregate power measurements. The extracted device profiles are then applied to disaggregate the aggregate power measurements by means of particle swarm optimization. Then, a new approach to very short-term power predictions is presented, which makes use of the disaggregation data. For this purpose, a state change forecast is carried out for each device by an artificial neural network and subsequently converted into a power prediction by reconstructing the power profile with respect to the state changes and device profiles. The forecast horizon is 15 minutes. In order to demonstrate the developed approaches, three-phase reactive and active aggregate power measurements of a multi-tenant commercial building are employed as a case study. The granularity of the data used is 1 s. In total, 52 device profiles are extracted from the aggregate power data. The disaggregation exhibited a highly accurate reconstruction of the measured power with an energy percentage error of approximately 1 %. The indirect power prediction method developed is then applied to the measured power data and outperforms the two persistence forecasts, as well as an artificial neural network designed for 24h-ahead power predictions working in the power domain.

Keywords: Non-intrusive load monitoring, energy disaggregation, power prediction, unsupervised learning, neural networks

1. Introduction

Due to a higher share of renewably generated power and the increasing electrification of our society, the electricity grid is facing a variety of new challenges such as instabilities arising from sudden increases in energy supply or demand. A possible solution to avoid overloading without massively increasing grid capacity is energy management applied to both the supply and the demand sides of the grid [1]. Energy management relies, among other things, on high-quality forecasts of electricity supply and demand for different time horizons, spanning seconds to months [2, 3, 4]. Horizon predictions at the scale of seconds to minutes are referred to as *very short-term* predictions, and are especially important for decision-making in energy management systems. Such predictions are carried out for multiple levels of the electricity grid, from high voltage grids to the device level [5, 6, 7]. Power predictions on the demand side must address the randomness of human behavior and thus exhibit erratic and highly volatile patterns. In particular, very short-term predictions are significantly influenced by randomness, and are more difficult to carry out than long-term ones [8]. The erratic behavior of power demand data is especially evident in households and industrial or commercial buildings. However, commercial buildings have great potential for carrying out high-quality power predictions, as they feature more repetitive demand than households due to the division of working time and non-working time by which the operate, for instance shift work and repetitive tasks. Additionally, commercial buildings generally have higher electricity demand than private households, as [9] shows to be the case in Germany. Thus, a single commer-

*Corresponding author, email adress: Stefan.Arens@dlr.de

**All authors were with: DLR Institute of Networked Energy Systems, Carl-von-Ossietzky-Str. 15, 26129 Oldenburg, Germany

cial building could exert a significant impact on the stability of the overall energy system .

In order to support energy management in buildings, non-intrusive load-monitoring (NILM) is employed to delineate the aggregate consumption data into the contributions of individual devices in a particular building. This was first described by Hart [10] and is also often referred to energy disaggregation. Most disaggregation methods, such as those presented in other studies, e.g. [12, 13], function by building a model based on prior knowledge and also training algorithms using labelled data sets, as in [14]. The data-sets used are mostly not available in real world applications, as their collectionrequires extensive metering infrastructures.

In general, households are mostly considered for NILM analyses [15]. These energy systems are very similar in terms of their components, which increases the algorithms transferability from one household to another. However, NILM remains laregely unexplored in the context of commercial buildings [15]. The results of NILM can be utilized for multiple purposes, as they generally yield additional insight into the respective building. In [16], the authors incorporate appliance usage patterns in order to improve of load-forecasting performance, and in [17] NILM is employed, as well as a subsequent clustering analysis of similarly functioning appliances as a preprocessing step in the development of a forecasting algorithm. Nevertheless, the knowledge and results generated by energy disaggregation are only seldom applied to power prediction tasks. In particular, to the best of our knowledge, the device state data has not been directly used for power predictions purposes. In addition, highly transferable NILM techniques, which are operative without any prior knowledge of a building or a costly pre-training, or which do not require a massive increase in metering infrastructure, remain largely absent. Therefore, it is difficult to apply NILM to real-world energy systems, especially in the case of commercial buildings, which have highly individual energy systems to which hardly any suitable NILM approaches are applicable [15], as pre-trained models and methods are do not work.

With respect to the power predictions, existing methods also face problems because they are often unable to predict sudden events that result in sharp power increases [8, 18, 19]. Even state-of-the-art artificial neural networks have difficulties in coping with these data structures [20, 18]. However, this event-like behavior is the inherent core of consumption data and highly important for obtaining high-quality power predictions.

This study integrates the fields of NILM with power forecasting. Herein, we rectify the lack of fully unsupervised NILM methods by presenting a new energy disaggregation approach based on statistical and unsupervised machine learning methods. The presented NILM approach calculates device profiles on the basis of aggregate power consumption data. The aggregate power is disaggregated by means of particle swarm optimization, as developed by the authors of this paper, which is extensively outlined in [21]. The main contribution of this work concerns the problems inherent to very short-term power predictions in buildings. For this purpose, a new bypass prediction method is presented that utilizes the state data of single devices based on the aforementioned NILM approach. We present a means of device state prediction and carry out a 15 min power prediction by reconstructing the aggregate power on the basis the state data derived from the corresponding device profiles. Hence, the event-like behavior is inherent to the state data and therefore highly suitable to very short-term power predictions. Figure 1 visually depicts the procedure we developed.

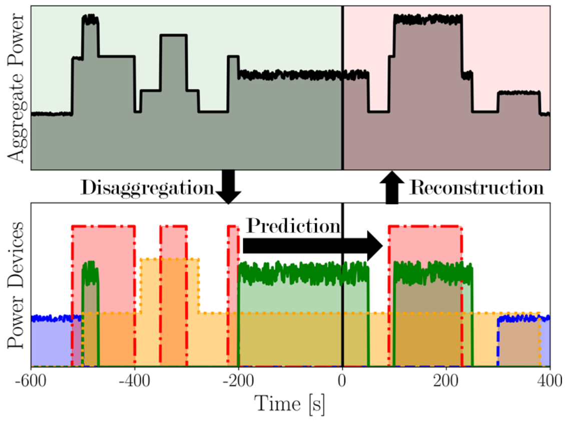


Figure 1: Graphical representation of power predictions based on power disaggregation. The aggregate power signal (top) is disaggregated until $t = 0$. This yields the power contributions of different devices (bottom). From $t = 0$, the state of the device is predicted and thereafter the aggregate power signal is reconstructed using the state data from single devices

This paper is structured as follows: In Section 2, we present the data set that is used. The methodology is then outlined in Section 3; we begin by describing the assumed disaggregation problem in Section 3.1. Afterwards, the developed and used methods are presented, including device profile extraction, disaggregation with particle swarm optimization and very short-term power prediction based on an artificial neural network. In order to highlight the results in a real world context, in Section 4 the developed methods are applied to the power data of a commercial building. Following a discussion and outlook in Section 5, we convey our conclusions in Section 6.

2. Data Description

In this study, we employ the power data of a single measuring point in a multi-tenant commercial building as a data set for our developed methods. The temporal granularity is 1 s. The data represents a production facility and workshop and contains six features: Three phases of active and three of reactive power, respectively. The six features of the power measurements are referred to as $P_0 \dots P_5$, with $P_0 \dots P_2$ representing the three active power phases and $P_3 \dots P_5$ the three reactive power phases, respectively. The data set encompasses power measurements from 1 December, 2018 to 29 March, 2019. On average 0.0023 % of data points are missing - these gaps are filled by the last known value. We utilize the UMG 604 PRO power analyzer from Janitza Electronics (Germany). According to the manufacturer the measuring error of this device is less than 0.4 % which we neglect herein [22]. Figure 2 displays the distribution of the data, that is used in this study.

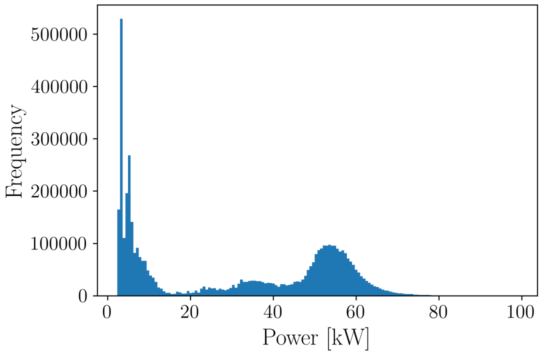
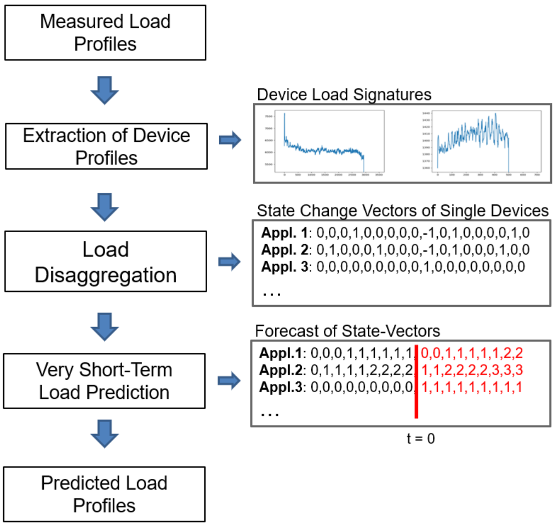


Figure 2: Histogram of the summed active power for the data used. The minimum is 2.26 kW, the mean 22.27 kW, and the maximum 98.95 kW.

3. Methodology

In the first of the sections that follow, the assumed formulation of the disaggregation problem is stated. Then, the device profile extraction method is presented with the particle swarm optimization (PSO) disaggregation method employed is then briefly described in the subsequent section. Finally the disaggregation-based power prediction procedure is outlined in the last of the following sections. In order to provide an overview of the algorithm, Figure 3 displays the major steps.



Measured Load Profiles
Extraction of Device Profiles
Device Load Signatures
Load Disaggregation
State Change Vectors of Single Devices
Appl. 1: 0,0,0,1,0,0,0,0,0,-1,0,1,0,0,0,0,1,0
Appl. 2: 0,1,0,0,0,1,0,0,0,-1,0,1,0,0,0,1,0,0
Appl. 3: 0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0
…
Very Short-Term Load Prediction
Forecast of State-Vectors
Appl.1: 0,0,0,1,1,1,1,1,1, 0,0,1,1,1,1,1,2,2
Appl.2: 0,1,1,1,1,2,2,2,2, 1,1,2,2,2,2,3,3,3
Appl.3: 0,0,0,0,0,0,0,0,0, 1,1,1,1,1,1,1,1,1
…
t = 0
Predicted Load Profiles

Figure 3: Graphical representation of the developed algorithm.

3.1. *Formulation of the Disaggregation Problem*

We assume a very similar formulation of the disaggregation problem to be described in [21]. The aggregate power at time $t \in \{0, 1, \dots, T\}$ termed $P(t) \in \mathbb{R}^6$, is assumed to be a linear combination of device profiles corresponding to their state changes, as described in the following equation [21]:

$$P(t) = \sum_{\substack{i,\tilde{t} \\ s_i(\tilde{t})=1}} s_i(\tilde{t}) l_i(t+\tilde{t}) + \sum_{\substack{i,\tilde{t} \\ s_i(\tilde{t})=-1}} s_i(\tilde{t}) \mathbb{1}_{(\tilde{t},T)}(t) p_i + \epsilon(t) \quad (1)$$

The device profile of device $i \in \{0, 1, \dots, M\}$ contains a dynamic profile l_i and a power value of the stable operating state $p_i \in \mathbb{R}^6$ with τ_i being the (typical) time required until this state is reached.

$S \in \{0, 1, -1\}^{T\times M}$ denotes the so-called state-changes-matrix with $s_i(t)$ being the t^{th} row and the i^{th} column of S. If $s_i(t) = 1$, device i is switched on at time t and for $s_i(t) = -1$ it is switched off. When $s_i(t) = 0$, the state of device i remains the same. $\epsilon(t)$ is referred to as an always-on component or noise. In light of these assumptions for the aggregate power signal, the following optimization problem must be solved [21]:

$$\min_{\mathrm{S}} E\Big(P, P_{\mathrm{S}}\Big) \tag{2}$$

P denotes the measured aggregate power signal, P_{S} represents the reconstructed or approximated power according to Equation 1 using the state changes matrix S and the device profiles l_i, and $E(P, P_{\mathrm{S}})$ represents an error function of P and P_{S}. The state changes matrix of S and the device profiles l_i must be identified in order to minimize the error E.

3.2. Extraction Procedure of Single Device Profiles

For device profile extraction, we assume a device with a binary state, i.e., the device can only be in an ON or OFF state. The stand-by modes or different operational modes of an appliance would be described as individual device profiles. This also applies to the complex programs of some appliances. Figure 4 presents a graphical and generic representation of the division of a complex appliance signature into a simplified set of device profiles.

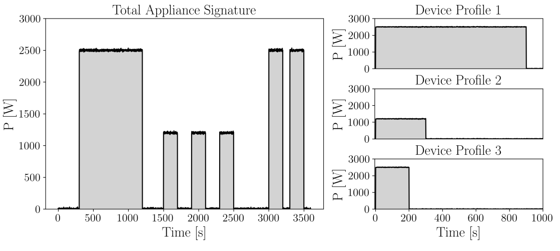


Figure 4: Graphical representation of the separation of complex appliance signatures into simple device profiles. The left profile contains repetitive patterns and is divided into three characteristic simpler profiles that represent such characteristic patterns.

The device profile extraction algorithm first detects the times of events in the aggregate power signal by identifying peaks in the derivative of the aggregate power signal. Then, the events are clustered using the k-means algorithm in order to determine the characteristics when switching the specific device types on or off. Subsequently, the clusters are cleaned and merged. In order to determine the typical run-time of the device, i.e., the length of its profile, the clusters are split using Gaussian mixture models (GMMs) according to the characteristic ON-duration. Finally, median blending is employed in order to extract the device profiles from the aggregate power signal.

3.2.1. Peak Analysis

We start by identifying when device state changes occur. For this purpose, we employ the derivative of the measured aggregate power signal P which is denoted by $\Delta P : \{1, \ldots, T-1\} \to \mathbb{R}^6$ and is calculated according to Equation 3 where $t + 1$ denotes the subsequently measured point in time with respect to t. Due to the constant measuring frequency of 1 Hz, the relationship is simplified to:

$$\Delta P(t) = \frac{P(t+1) - P(t)}{(t+1) - t} = \frac{P(t+1) - P(t)}{1\,\mathrm{s}} \tag{3}$$

We assume that a state change occur when a sharp increase or decrease in the measured power is observable. These inflection points in the aggregate power signal result in maxima or minima in the derivative. In the following, maxima are referred to as ON events and minima as OFF ones. In order to identify events, we take the sum of active phases, $P_{\text{tot}} \in \mathbb{R}^T$ with $P_{\text{tot}} = P_1 + P_2 + P_3$, into account. We perform a peak analysis of the derivative of the sum of the three phases of active power ΔP_{tot} with $\Delta P_{\text{tot}} \in \mathbb{R}^{T-1}$. For the peak analysis, we take all of the values of ΔP_{tot} into account, which are above a threshold value $\varepsilon_{\text{threshold}}$, and so $|\Delta P_{\text{tot}}(t)| \geq \varepsilon_{\text{threshold}}$. The threshold can be chosen on the basis of the given power data. The process of selecting a peak threshold could be automated in the future. We assume that the process of switching a device on or off is completed within 1 sec. When $\Delta P_{\text{tot}}(t)$ is an event, we denote the respective time by t_{p} and term t_{p} the event-time. We introduce the following peak criterion, which defines time t as an ON-event time t_{p}:

$$\begin{aligned} t = t_{\text{p}} \Leftrightarrow \Delta P_{\text{tot}}(t-1) < \Delta P_{\text{tot}}(t) \wedge \\ \Delta P_{\text{tot}}(t+1) < \Delta P_{\text{tot}}(t) \\ \wedge\, \Delta P_{\text{tot}}(t) \geq \varepsilon_{\text{threshold}} \end{aligned} \tag{4}$$

Equation 4 accordingly applies for OFF-events with reversed signs. The set of N events is referred to as $D = \{\Delta P(t_{\mathrm{p},1}), \ldots, \Delta P(t_{\mathrm{p},N})\}$.

3.2.2. Cluster Analysis of Events

The relationship between active and reactive power is found to be distinctive for specific types of devices [23]. Therefore, we assume the increase or decrease in the three phases of active and reactive power at the time of an event to be characteristic of the particular device type. Based on this assumption, we can cluster the extracted events $\Delta P_{\mathrm{tot}}(t_{\mathrm{p}})$ according to their characteristics in all six power features in order to distinguish the device types. For the cluster analysis, we employ the well-known k-means cluster algorithm. It is assumed that the specific patterns of an ON-event correspond to those of an OFF-event with reversed signs. The clustering analysis is therefore only performed for the ON-events, with the OFF-events assigned to the cluster centers reversed signs that feature the smallest subsequent deviation . The k-means cluster algorithm divides a given data set $D = \{\Delta P(t_{\mathrm{p},1}), \ldots, \Delta P(t_{\mathrm{p},N})\}$ into K clusters in such a way that the Euclidean distance of each data point to the nearest cluster center is minimized. The number of clusters K must also be given. This can be formalized as:

$$\min_{r_{nk},\vec{c}_k} \quad \sum_{n=1}^{N}\sum_{k=1}^{K} r_{nk}|\Delta P(t_{\mathrm{p},n}) - \vec{c}_k|^2 \tag{5}$$

$r_{nk} = 1$ if the event $\Delta P(t_{\mathrm{p},n})$ belongs to cluster k and $r_{nk} = 0$ for all other clusters. The cluster centers are denoted by $\vec{c}_k \in \mathbb{R}^6$ and the corresponding cluster constitutes a set of assigned events as denoted by c_k. The k-means cluster algorithm solves the minimization problem by means of the expectation-maximization method [25]. In order to determine the optimal number of clusters K_{opt} for the given events, the Calinski-Harabasz (CH) score is used, which is defined by [26]:

$$\mathrm{CH} = \frac{N-K}{K-1}\frac{\sum_{c_k\in C}|c_k||\vec{c}_k - \vec{D}|^2}{\sum_{\vec{c_k}}\sum_{\Delta P(t_{\mathrm{p},i})\in c_k}|\Delta P(t_{\mathrm{p},i}) - \vec{c_k}|^2} \tag{6}$$

where N denotes the number of events, the center of the entire data set D is denoted by $\vec{D}$ and C represents the set of clusters c_k. The cardinality of cluster k is denoted by $|c_k|$. CH reaches a maximum for the optimal K and calculates a ratio between the separation of the clusters and the compactness within each of them. It is then multiplied by the pre-factor $\frac{N-K}{K-1}$ in order to prevent overfitting, because a larger number of clusters K must not always result in a higher value of CH than a smaller number of clusters.

In order to obtain K_{opt}, we perform a k-means clustering analysis for $K \in \{1 \ldots 50\}$ and calculate CH each time. We selected 50 as the upper limit in order to confine the computing time. An adaptive method for increaseing K until the CH is decreasing again would also be possible.

Following the first clustering analysis of the extracted events, we perform a cleaning step of the clusters, analogous to that reported in [23]. For this, we define outlier events $\Delta\tilde{P}(t_{\mathrm{p}})$ to be outside of a 2σ area within the respective cluster, where σ denotes its standard deviation. All of the outliers are clustered again with a fixed $\tilde{K} = 10$. A second CH-analysis would also be possible for the outlier events, but this step is simplified as this cleaning step is optional in the procedure for extracting the device profiles. Using the presented clustering procedure, the characteristic increase or decrease in all six power features when switching a device on or off is known.

3.2.3. Merging Clusters

In order to improve the clustering of the extracted events, we perform a merging step of clusters based on a similarity measure. The similarity of two clusters is evaluated by means of the Pearson correlation coefficient $\rho \in [-1, 1]$ and the absolute percentage error (APE) calculated for each combination of two cluster centers. The Pearson correlation coefficient of the two cluster centers $\vec{c}_i$ and $\vec{c}_j$ is defined by the following equation [27]:

$$\rho(\vec{c}_i, \vec{c}_j) = \frac{\sigma_{\vec{c}_i,\vec{c}_j}}{\sigma_{\vec{c}_i}\sigma_{\vec{c}_j}} \tag{7}$$

where $\sigma_{\vec{c}_i,\vec{c}_j}$ denotes the co-variance of $\vec{c}_i$ and $\vec{c}_j$. The APE is defined by the following equation:

$$\mathrm{APE}(\vec{c}_i, \vec{c}_j) = \frac{|\vec{c}_i - \vec{c}_j|}{|\vec{c}_i|} \tag{8}$$

If $\rho(\vec{c}_i, \vec{c}_j)$ and $\mathrm{APE}(\vec{c}_i, \vec{c}_j)$ are above/below a given threshold, clusters i and j are merged. For this, a new cluster is created and the members of clusters i and j are assigned to i with cluster center $\vec{c}_{i,\mathrm{new}} = 1/2 \cdot (\vec{c}_i + \vec{c}_j)$. This calculation of the new cluster center is also applied if the cardinalities of c_i and c_j differ.

The following thresholds are selected:

$$\rho(\vec{c}_i, \vec{c}_j) > 0.9 \quad \wedge \quad APE(\vec{c}_i, \vec{c}_j) < 0.1 \tag{9}$$

It is possible that cluster c_i satisfies the condition in Equation 9 with multiple other clusters. If that scenario applies, c_i is only merged with the cluster of highest similarity; the newly created cluster $c_{i,\text{new}}$ is not merged again with the other clusters.

3.2.4. Determination of Run-Time with GMM

For the calculation of device the profiles, the typical run-time, i.e., time in the ON state, is required. Therefore, we determine the time between an ON-event in a specific cluster and the next OFF-event in that cluster for each of its ON-event. We perform this calculation for every cluster of events. The calculated time is referred to in the following as the ON-duration. If there are more ON- than OFF-events, we neglect these surplus ON-events and vice-versa. For every cluster, we present all of the determined ON-durations in a frequency distribution and observe multiple maxima at different time points.

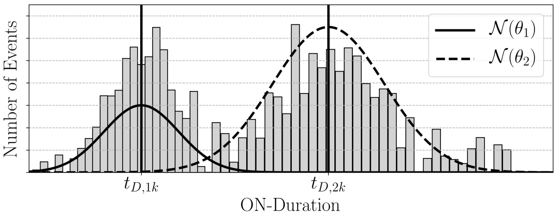


Figure 5: Graphical representation of the division of an ON duration distribution based on GMM.

In reality, the ON-duration depends on the type of use of the individual device,e.g., if a device is capable of running different programs or if the same device type is used for different tasks. In this study, GMMs are used to divide clusters according to their characteristic ON-durations. Figure 5 shows an exemplary distribution of the ON-duration distribution of a cluster with a fitted GMM that divides the distribution into two sub-distributions.

GMMs determine the properties of the sub-distributions within an overall distribution, based on only observations of the overall distribution $B = (\vec{x}_1, \ldots, \vec{x}_N)$ [28]. The a-posteriori probability for the GMM is calculated as follows:

$$p(\theta|B) = \sum_{i=1}^{m} \pi_i \mathcal{N}(\vec{x}|\vec{\mu}_i, \Sigma_i) \tag{10}$$

where $p(\theta|B)$ describes the probability of the model parameters θ given the data set B. The parameters $\theta_i = (\pi_i, \vec{\mu}_i, \Sigma_i)$ denote the mixing coefficients, the mean value,s and the covariance matrices of the i^{th} of the m Gaussian distributions. The mean values of the Gaussian distributions represent the mean ON-duration, which will be referred to hereinafter as d. Therefore, the ON-duration of device i is denoted by d_i. The number of sub-distributions m must be given beforehand. The *maximum-likelihood* method is used, together with the *expectation-maximization* algorithm to obtain an optimal estimation of θ [29]. In order to determine the optimal number of Gaussian distributions m_{opt} in the GMM of each cluster, the Bayesian information criterion (BIC) is used. The BIC is a measure for comparing different models and is defined by the following equation [30]:

$$\text{BIC} \approx \frac{1}{2} M \ln N - \ln p(B|\theta) \tag{11}$$

where N denotes the number of data points in the data set, and B and M are the number of parameters in θ. According to this definition, the BIC is to be minimized. As soon as $\Delta\,\text{BIC} > 2$ for two subsequent models $\mathcal{M}_m$ and $\mathcal{M}_{m+1}$, the model $\mathcal{M}_{m+1}$ is selected and the corresponding m is termed m_{opt}. The limit for $\Delta\,\text{BIC}$ to select m_{opt} must be empirically determined . In general, m should be increased as long as $\Delta\,\text{BIC}$ is negative for two subsequent models.

Given this procedure, every cluster k is divided into m *groups*. The groups that emerge from one cluster share its center (the characteristics of ON- and OFF-events) but differ in their characteristic ON-duration. An event is assigned to a group if the associated Gaussian distribution is the maximum for the ON-duration of this event. From the total K clusters emerge $M = \sum_{k=1}^{K} m_{\text{opt},k}$ groups, which will be denoted as G_i. The ON-duration of G_i is referred to as d_i.

3.2.5. Median Blending

For the final calculation of the device profiles, median blending, a method of noise reduction, is used for all groups [31]. We sought to extract the load profile of a single device type from the aggregated load profile by utilizing the previously determined activation peaks and ON-duration. Thus, all other devices that are not under consideration apply noise to the time series, which is in turn eliminated by median blending. For every element $\Delta P(t_{\text{p}}) \in G_i$, we store and normalize the aggregate power signal, which is denoted by P_{norm}, from $t_{\mathrm{p}} \dots t_{\mathrm{p}} + d_i$. Normalization is carried out by dividing the aggregate power during activation of a device type by the maximum power value within the stored segment of the aggregate power signal. Then, the median for every time point in the saved aggregate power signal is calculated for all six power features. In order to scale the normalized profile back to absolute power values, we employ the cluster center of the respective cluster, which represents the characteristic increase in power per second when switching on a specific device type. Therefore, we integrate the cluster center by multiplying it by one second. Finally, we scale back the median values by multiplying $\vec{c}_k$ and the normalized l_i. We define the power profile of device i, thus:

$$l_i : \{1, \dots, d_i\} \to \mathbb{R}^6 \,, \tag{12}$$

where d_i denotes the ON-duration, by:

$$\begin{aligned} l_i(t) = \vec{c}_k \cdot \mathrm{median}\{&P_{\mathrm{norm}}(t_{\mathrm{p}} + t) | \\ &t_{\mathrm{p}} \text{ is an ON-event of the device}\} \end{aligned} \tag{13}$$

for every $t \in \{1, \dots, d_i\}$. The prerequisite for this procedure are enough events in G_i to significantly reduce the noise of the aggregate power signal.

3.3. Disaggregation Procedure

The disaggregation is carried out by means of a PSO as described in a previous study of the authors of this paper [21], which is an improved version of the original description by Kennedy and Eberhart in [11]. The PSO is a metaheuristic that is used for multidimensional optimization problems, such as the above presented disaggregation problem. In this study, we employ PSO to determine the state changes of matrix S. For this purpose, the extracted device profiles are used. The PSO is intended to minimize the following error measure [21]:

$$\begin{aligned} E^{[a,b)}(P, P_{\mathrm{S}}) = \alpha \cdot &\sum_{t=a}^{b-1} (\vec{P}_{\mathrm{S}}(t) - \vec{P}(t))^2 + \\ \beta \cdot &\sum_{t=a}^{b-2} (\Delta\vec{P}_{\mathrm{S}}(t) - \Delta\vec{P}(t))^2 \end{aligned} \tag{14}$$

with $\alpha + \beta = 1$ weighting the two summands. The algorithm we employ in this study to carry out the disaggregation is extensively outlined in [21]. In that work, it is assumed that a device profile consists of transient or dynamic behavior and a stable state reach after a specific time τ. Thus, we assume that the extracted load profiles represent the dynamic behavior of the device. The power value of the stable state is assumed to be the final non-zero power value of the specific device profile.

3.4. Very Short-Term Power Prediction

This work presents a novel load disaggregation-based power forecasting methodology for estimating the state changes of unknown devices. The power forecast is carried out by reconstructing the state changes forecast according to Equation 1. In the dataset used, the weekends feature very regular power curves with highly repetitive patterns. Thus, it is assumed that persistence forecasts will be sufficient for weekend days. Therefore, we only consider working days, since they exhibit more complex power demands with many sharp increases and decreases. For this purpose, we utilize an artificial neural network (ANN). ANNs have been widely used in different fields pertaining to power grids as a very powerful method for time series prediction [32, 33, 34]. In particular for load and energy forecasts, ANNs are preferred due to the non-linearity and randomness that characterize power data [8]. In this study, we design the ANN to learn the interrelationship between the last hour of state changes and those to come within the next 15 minutes. We employ a feed forward, fully connected ANN based on the supported models and functions of *Keras* [35], a deep learning framework for building ANNs. In the following, the ANN input and output are described, as well as the hyperparameter optimization of the ANN. Finally, the training procedure for the ANN is outlined.

3.4.1. ANN Input and Output

This section describes how the state changes, optianed from the disaggreagtion precedure, are fed into an ANN. Additionally, the ANN's output is described. All inputs and outputs must be normalized to a range of $-1 \dots 1$, so that no feature is weighted more than any other during training. We utilize past data on the state changes of every device type as inputs and future state changes information as the target data.

The state changes of the previous 3600 s are used to predict the next 900 s, also considering the state changes for the respective prediction time from one week ago. For example, in order to predict the state changes from 11:00-11:15 am, the state change data from 10:00-11:00 am is required as an input from the respective day and, furthermore, the state changes from 11:00-11:15 from one week before are required. This arrangement has proven to be helpful given the regularities in industrial and commercial data relating to weekdays [36].

During training, the difference between the output and target data is quantified by calculating an error measure. A large proportion of the state change data consists of zeros. Thus, there is a local optimum in the error measure of the ANN to only predict zeros, resulting in no state changes at all. Therefore, we perform an additional preprocessing step for the state changes data by transforming the state changes data into state data by means of integration. Figure 6 displays a graphical representation of the integration procedure of the state changes. Essentially, the state changes are added up for a respective device. This step avoids a data structure that primarily contain zeros, caused by state changes that rarely take place. Following integration, the state data is normalized to the range of $[-1 \ldots 1]$, because a device could be activated several times, meaning that several devices of the same type are running, resulting in the state data containing values greater than 1. The integration step applies to both the input and the target data. This transformation exhibited an improved learning process of the ANN during training in comparison to the learning of the state changes data.

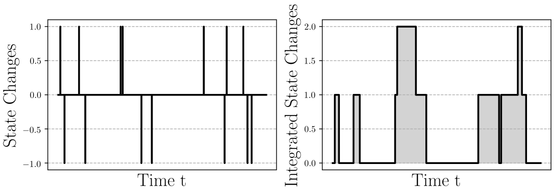


Figure 6: Graphical representation of the integration of the state changes to states for the data preparation for the training of the ANN.

Additionally, we introduce three time features as inputs: The first two are the sine and cosine functions, as presented in the left and middle panels of Figure 7. The third time features represents the day of the week: We assign a value to each day from Monday (0) to Friday (1), as is shown in the right pannel in Figure 7.

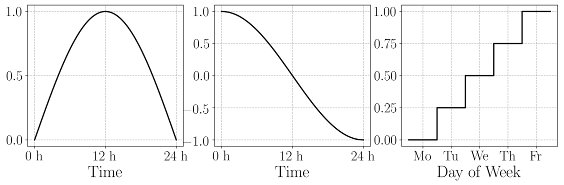


Figure 7: Graphical representation of the time features given the ANN as the input.

For M given device types, the input data set contains $2M + 3$ columns. The target data only contains the future state changes data of the M devices. Thus, there are M columns in the target data set. The number of rows is determined by the size of the training data set, and so this corresponds to the number of time steps in the training data set.

3.4.2. Hyperparameter Optimization

The hyperparameter optimization was carried out with the help of *Talos* and the supported random search [37]. Talos is a library tailored for the hyperparameter optimization of *Keras* models. Hyperparameters comprise all of the parameters of an ANN that are not adapted during training but must be set beforehand. The following hyperparameters are considered for the optimization: *Number of neurons*, *number of hidden layers*, *dropout*, *learning rate*, and *batch size*. The *number of neurons* in the hidden layers sets the width of the ANN, whereas the *number of hidden layer* determines the ANN's depth. The number of neurons in the hidden layers are not the same for all layers, and therefore the width of the ANN can vary. *Dropout* describes the percentage of neurons that are randomly neglected in every hidden layer during a training step in order to increase the ANN's robustness and decrease over-fitting [38]. The *learning rate* is a measure of the step size made during the training process in each iteration. A larger learning rate decreases the training time, but increases the risk of not fully converging into an error minimum and vice-versa for smaller learning rates. The *batch size* determines the number of samples of the training data set that are simultaneously processed. Thus, the parameters of the ANN are not adapted after every single sample passes it, but only after the number of samples corresponding to the batch size.

The chosen values of the optimized hyperparameters are presented in Table 1. In light of these, the chosen model has 137868 trainable parameters. As an activation function, the *relu* function proved to have the best outcome for our objective.

Table 1: Hyperparameters and their chosen values for the state forecast using an ANN.

Hyperparameter Name	Selected Value
Neurons in hidden layer	214
Number of hidden layers	3
Dropout	5 %
Learning rate	0.01
Batch size	2048

We select the mean squared logarithmic error (MSLE) as the error measure which is defined as follows:

$$\mathrm{MSLE}(y_{\mathrm{target}}, y_{\mathrm{Out}}) = \frac{1}{N}\sum_{i=0}^{N}\left(\frac{\log(y_{\mathrm{target},i}+1)}{\log(y_{\mathrm{Out},i}+1)}\right)^2 \quad (15)$$

Due to its logarithmic character, the MSLE penalizes deviations at small values more heavily than error measures such as root mean squared or mean absolute formulas. This demonstrated an improved training process given the structure of the data herein. In order to evaluate and compare the results of the prediction, we use the same error measures as for the validation of the disaggregation results.

3.4.3. Training of the Neural Network

The training is performed using an Intel i7-6700k processor with 16GB of RAM and a Geforce GTX 1050 graphic card with 768 CUDA cores. The data set for the training includes 55 working days from January-March 2019, and so it contains 4752000 rows and $2M + 3$ columns. During training, 95% of the data is used for training the network and 5% get as a validation data set. Note that, due to the data's high measurement frequency, the test set still consist of 172800 data points, which result in 188 very short-term predictions to be tested. As soon as the error on the independent validation set increases, the training is stopped. This is carried out using the early stopping option of *Keras* [35]. As a postprocessing step, we calculate the derivative, and so the reverse procedure of the shown integration is applied. The output values of the ANN used are floats rather than integers, as assumed in Equation 1. Thus, we interpret the outputs as probabilities of the state changes of the devices. In order to reconstruct the power, we allow the floats and calculate a weighted sum rather than a discrete one. Therefore, Equation 1 changes as follows:

$$P(t) = \sum_{\substack{i,\tilde{t}\\ s_i(\tilde{t})>0.1}} s_i(\tilde{t}) l_i(t+\tilde{t}) + \sum_{\substack{i,\tilde{t}\\ s_i(\tilde{t})<-0.1}} s_i(\tilde{t}) \mathbb{1}_{(\tilde{t},T)}(t) p_i + \epsilon(t) \quad (16)$$

with $s_i(t) \in \mathbb{R}$. We define a threshold of 0.1 to take an element of the prediction into account for the purpose of reconstruction. As the always-on-component ϵ, we assign each short-term prediction the last measured power value. Therefore, for a prediction in the range from $t = 0$ to $t = 900$, we set $\epsilon = P(-1)$. The forecast is limited to being greater than zero for active power values, as negative active power values are not possible assuming only that consumers are connected.

All operations and associated data are depicted in step-by-step fashion in Figure 8.

4. Results

In this section, we present the results of the application of the developed methods to the above described data set. For the device profile extraction, we employ the data from January and February of 2019. Thereafter, we disaggregate the entire data set. In order to train the forecast algorithm, we use the data from January to March 2019. The testing of the forecast algorithm is carried out using the last two days in the data set, namely 28th and 29th of March, 2019. As the forecast horizon is 15 min, we are able to perform and evaluate 188 power predictions of the test data set. In order to validate the results of the short-term prediction, we use the root mean squared error (RMSE), mean absolute error (MAE), mean absolute percentage error (MAPE) and percentage energy difference ($\mathrm{Energy}_{\mathrm{E}}$) error measures. Moreover, the RMSE and MAE are used to compare the short-term prediction with a day-ahead prediction.

In Figure 9, an exemplary cluster analysis of one day of data (December 4th, 2018) is shown for the elements: $\Delta P_2(t_{\mathrm{p}})$ and $\Delta P_5(t_{\mathrm{p}})$. The ON-events are depicted in the right-hand panel of Figure 9, with the OFF-events in the left-hand one. The symmetry to the central point zero is clearly visible by comparing the the left- and right-hand sides of

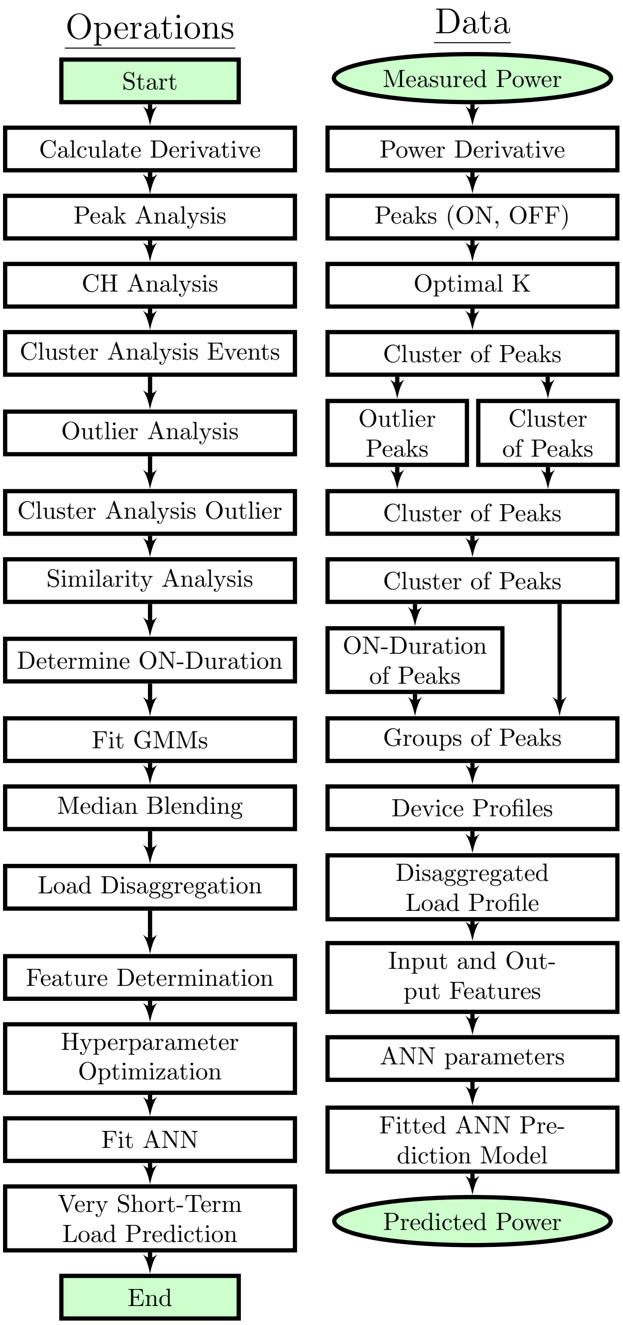


Figure 8: Graphical representation of the developed algorithm. On the left, the operations are depicted, whereas the data of every step of the algorithm is displayed on the right.

Figure 9. It is apparent that the relation of the increase in active and reactive power is not randomly distributed, but forms clusters, with ON- and OFF-events being individually clustered . The cluster-forming behavior becomes clearer when taking all six features of the power derivative into account. Therefore, all six features of the six exemplary cluster centers are presented in Figure 10, where it can be seen that the clusters have very distinct characteristics with respect to the relationship between the six features. Whereas Examples 1, 3 and 5 only show an increase in one phase of power, whereas the other three examples appear to represent three-phase connected devices. They have approximately the same power derivative during an ON-event in all three phases with respect to active and reactive power. The relationship between active and reactive power is very distinct. Whereas examples 1, 3, and 5 exhibit almost no increase in the reactive power when switched on, examples 4 and 6 feature significant reactive power increases. In Example 4, the increase in reactive power is even higher than the increase in active power.

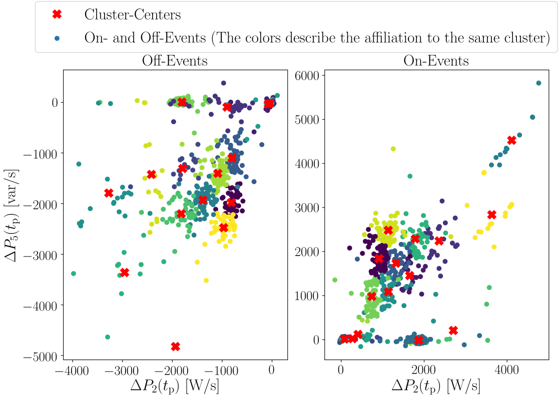


Figure 9: Individually clustering of OFF-events (left-hand side) and ON-events (right-hand side) for December 4th, 2018. Two of the six power derivative features are presented.

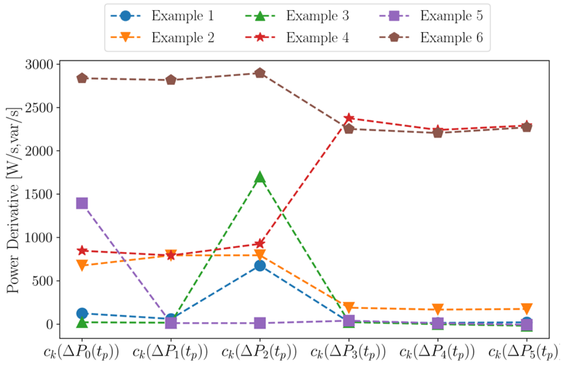


Figure 10: Six examples of cluster centers and their characteristics in all features of the power derivative.

In order to show the separation of clusters according to their ON-duration, Figure 11 displays two exemplary ON-duration distributions with the respective fitted GMMs. Cluster 15 from Figure 11 is divided into two groups with approximate ON-durations of 200 s and 1000 s. On the other hand, Cluster 14 is divided into three groups with approximate ON-durations of 250 s, 900 s and 1900 s.

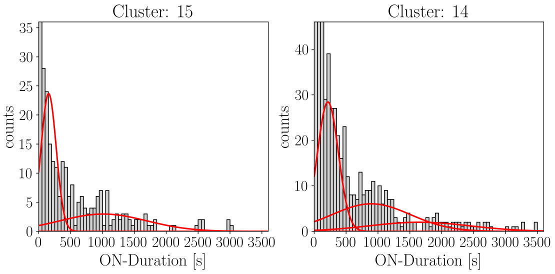


Figure 11: Two examples of GMMs for ON-duration distributions

Given the examples for different steps of the developed algorithm, Figure 12 shows four device profiles. In total, we extracted 52 device profile from the aggregate power data using the developed algorithm. The depicted profiles are representative for all extracted device profiles, as they show the main patterns and behaviors of the extracted device profiles . Both upper illustrations in Figure 12 display the most common type of device profile: A three-phase connected device with transient behavior in the beginning and afterwards a stable operating state in which the relationship between active and reactive power remains approximately the same. Additionally, device profiles 4 and 42 indicate that the relationship of active and reactive power is characteristic to the specific device. The length of profiles 4 and 42 also differ. Profile 6 exhibits no dynamic behavior at the beginning of the profile, but consists of a constant component and an oscillating or random one. Clearly, Device Profile 6 represents a single-phase connected device, as all power values other than P_1 are close to zero. Moreover, Profile 6 has a high ON-duration compared to profiles 4 and 42, with $d_6 \approx 20000\ s$. An exception to the frequent device profile pattern is represented by Device 37, which shows a decreasing behavior with many small but sharp increases and decreases in all power features. These kinds of device profiles are less common.

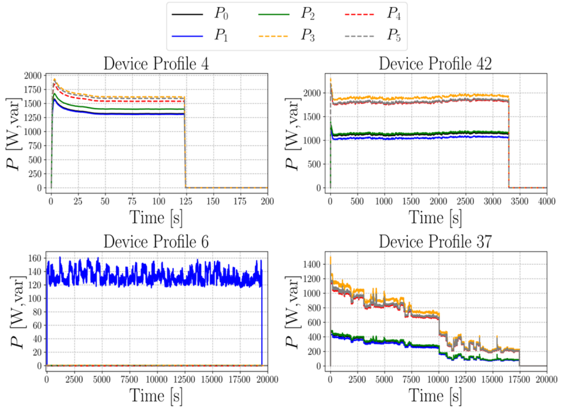


Figure 12: Four examples of device profiles extracted in an unsupervised manner from the aggregate power signal. The solid lines represent active power and the dashed lines reactive power.

4.1. Disaggregation

In total, we disaggregated the power data of 119 days from December 2018 to March 2019. Figure 13 shows a typical day of data with the sum of active phases on the left and the sum of reactive phases on the right. Beneath these, the respective absolute error is displayed. It can be seen that the PSO is able to reconstruct the shape of the aggregate power signal over the duration of a entire day, including repetitive patterns during the night and across most of the peaks. Nevertheless, there are error peaks of up to almost 20 kW, which correspond to approximately 25 % of the measured power at the corresponding time points. However, these high error values occur infrequently and are of very short duration. During working periods, the absolute error is higher than at night, but there is no constant offset between the measured and reconstructed power. The error of reconstructing the reactive power is larger than that of reconstructing the active power. At the end of the presented day, noise is present in the reconstructed power.

For the working days in March 2019, the RMSE is 1565 ± 150 W and the energy error is 0.897 ± 0.156 % between the reconstructed time series after disaggregation and the original, measured time series [21]. Note that the days are individually evaluated . Therefore, the standard error describes the variation between the different days. The reproducibility of the results is demonstrated by the standard deviations of the mean error values which are approximately 10% of the respective mean values.

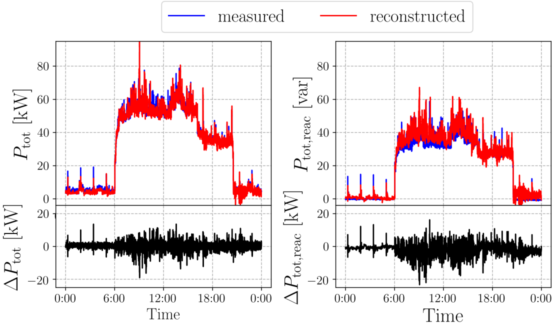


Figure 13: Disaggregation results for 4th December, 2018. On the left is shown the sum of the active power shown and on the right side that of the reactive power . At the bottom is illustrated, the respective absolute difference between the measured and reconstructed power.

4.2. Short-Term Power Predictions

Drawing on the data produced by the disaggregation, an ANN is trained according to the description in Section 3 of the data's pre- and postprocessing procedures. The data set used for testing the ANN's performance consists of the 28th and 29th of March, 2019. Therefore, using this test dataset, we can calculate 188 power predictions of 15 minutes each. The ANN is given the first hour of the test set as input data. To put the results into perspective, we compare the error measures on the test set with those for different persistence forecasts.

All of the error measures are calculated for the sum of the active power phases. Table 2 shows the means and standard deviation of multiple error measures for the predictions using two persistence methods for comparison. The first persistence forecast utilizes the power values from seven days prior, whereas the second uses the power values of the preceding 15 min. The ANN outperforms both of these with respect to mean error values of all of the calculated error measures. In particular, the MAPE and error in daily consumed energy are significantly smaller.

We compare the developed short-term prediction based on state change data with the prediction results of an ANN based mainly on past power data with a granularity of 5 min, adapted from [36]. In that study, the authors employed the same data as in this work and optimized a long-short-term-memory neural network for a 24 h day-ahead prediction. Although the prediction horizon and granularity differ, the power prediction from [36] represents the standard prediction procedure and therefore serves as a benchmark . The model from [36] is used to predict the power for the test set herein. Figure 14 shows the measured power and both ANN predictions.

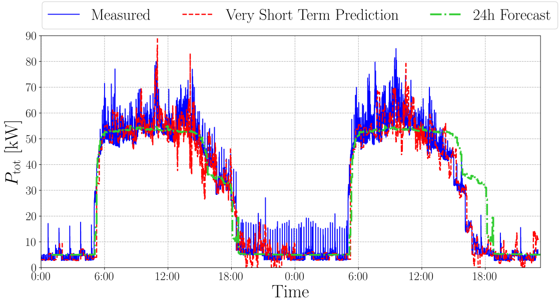


Figure 14: Comparison of the 24 h day-ahead power forecast and very short-term power prediction based on the state forecasts

The 24 h day-ahead prediction is similar to a rolling averaged power value, whereas the short-term prediction based on state change data exhibits the erratic behavior during working time, with sharp increases and decreases in the power level. With the model of the 24 h day-ahead prediction, we can calculate the RMSE and MAE. Table 3 shows the results of this. The RMSE and MAE are significantly higher in the 24 h day-ahead predictions than in the short-term prediction using the disaggregation-based ANN.

5. Discussion and Outlook

For the extraction of the device profiles, the main distinguishing factor is the behavior at an ON-event. The peak criterion used to determine the ON-behavior is very simple and neglects peaks across multiple time-steps. This problem could be resolved in future work using a more sophisticated peak criterion.

The k-means clustering algorithm is used to the determine clusters in the six-dimensional space of reactive and active power phases. Furthermore, other studies have used clustering to differentiate between device types, e.g. [23, 24], but these use a maximum of two features and not six, as in this work. In general, clustering is more precise, when more characteristic features are present [25]. Thus, we can assume that we can reach a higher degree of precision in dividing the events into clusters of device types. Other properties measured by power analyzers could also be utilized in future work to

Table 2: Multiple error measures between the measured and predicted power for two different persistence forecasts and the proposed algorithm, which makes use of states fed into an ANN. Presented are the means and standard deviations of the errors. These are calculated for 188 individual predictions of 15 minutes each for the test data set 28th - 29th March, 2019.

	Persistence 7 days before	Persistence 15 min before	Power prediction with ANN
RMSE [W]	6148 ± 5755	4327 ± 4045	3478 ± 3444
MAE [W]	5370 ± 5762	3379 ± 3722	2693 ± 3271
MAPE [%]	78.06 ± 117.84	20.17 ± 21.45	16.56 ± 20.19
Energy_E[%]	35.25 ± 119.10	3.40 ± 24.19	3.40± 24.19

Table 3: Multiple error measures between measured and predicted power for a 24 h day-ahead prediction and the proposed short-term prediction using states and an ANN.

	24 h day ahead prediction	Power prediction with ANN
RMSE [W]	5124	3478
MAE [W]	4507	2693

distinguish between different device types. Nevertheless, the number of necessary features should be limited with respect to realistic applications in real-world energy management systems and the availability of high-resolution power analyzers.

During clustering, we assume that the cluster centers of OFF-events are the reversed cluster centers of ON-ones. When clustering is performed for ON- and OFF-events individually, the OFF-event cluster centers, where reversed signs lie within a 0.25-σ area of the ON-event cluster center with σ denotes the standard deviation of the corresponding cluster. Therefore, the assumption can be justified. Additionally, Figure 10 displays the symmetry of the central points of the ON- and OFF-events.

In order to determine the ON-event behavior, we perform a peak analysis based on the assumption, that the switching procedure of a device finished within one second. In reality, most devices display a transient behavior in the shape of exponentially decreasing oscillation [39]. Some researchers have distinguished between different transient behaviors and so different device types [40, 41]. However, with respect to the measuring frequency, these processes occur on shorter timescales (within milliseconds) and can be neglected here. Only with a measuring frequency in the range of kHz would the characteristic transient behavior be observable [39]. However, the installation of an infrastructure that is able to perform measurements in kHz is unlikely. Thus, the presented approach, using a measuring frequency of 1 Hz, is more realistic for application to local energy and power management systems. With the measuring frequency of 1 Hz utilized in this study an ON-event approximates a step function in the aggregate power signal. Nevertheless, in most of the device profiles shown in Figure 12, transient behavior can be observed in the first few seconds of the corresponding profiles. Thereafter, most devices reach a stable state where they remain for the duration of the profile. for as long as the profile persists. Thus, the division of the device profiles into stable state and dynamic behaviors for the stated formulation of the disaggregation problem can be applied here.

The final step in the device profile extraction procedure is median blending. In general, more samples for performing median blending will yield more precise results. Therefore, it is very important to perform the extraction procedure using a sufficient amount of data. In particular, devices that are only rarely switched on provide less accurate profiles. The selected normalization is carried out by means of a division by the maximum power value in every sample of P_{norm}. With a high base load, this procedure could average out the characteristic fluctuations of the device profile. Therefore, another normalization method could be appropriate if the individual profiles are of great importance and an allocation to real measured profiles is of interest. However, in this study we focused on high-quality very short-term power predictions with an emphasis on the aggregate power signal. Thus, small fluctuations in individual device profiles were of minimal importance. The improvement in the median blending procedure or the application of other noise-reduction methods for device profile extraction could be examined in future studies.

In total, 52 device profiles are extracted for our dataset of a commercial consumer. As no additional information about the used data is available, we can not validate the number of device profiles. However, we can estimate this relatively high number of device profiles through the division by ON-behavior, as well as the division of clusters into groups with similar run-times. Even a simple ohmic consumer type could therefore result in multiple device profiles.

A direct validation of the device profiles was not possible in this study due to a lack of data on the correct device profiles. Moreover, the application of the extracted device profiles to the measurement of complex appliance signatures would be difficult, as the extracted profiles only represent the operational modes of appliances. However, the good results in the disaggregation and forecast indicate that the extracted device profiles are a satisfactory representation of the real devices.

The extraction procedure has similarities to non-negative blind sources separation in acoustics, where the individual components and mixing procedure are unknown [42]. As all of the methods used for extracting the device profile are from statistics and unsupervised machine learning, no hyperparameters have to be optimized to apply the algorithm to different data sets. The required hyperparameters, such as the number of K clusters or the number of Gaussian distributions in the GMMs, are determined using statistical scores or criteria. Therefore, the device profile extraction algorithm can be applied without changes. The suitability for transferability must be systematically examined in the future.

Figure 13 shows that the disaggregation discussed in this work can achieve a highly accurate reconstruction of the measured power. The results are consistently good across all six phases. Thus, we can assume that the device profiles constitute a good representation of the real devices, and also that the separation according to the ON-event behavior appears to be valid. As the PSO is a metaheuristic, incorrect assignments of devices to events are possible. Nevertheless, the disaggregation procedure produces additional knowledge of the building or the respective data set without requiring a costly model and its adaptation to the data. The aim of this work is the use of this additional knowledge for the purpose of very short-term power prediction and to determine if this additional knowledge provides benefits for such applications. The disaggregation procedure can be justified if a disaggregation-based prediction method is able to outperform standard prediction methods utilized in the power domain.

The very short-term forecast conducted using state changes data exhibits significantly better results than multiple persistence forecasts and a forecast using a LSTM network that is optimized for 24 h prediction with a resolution of 5 min. Thus, the LSTM predicts 288 values for an entire day compared to the 900 of our short term forecast for 15 minutes. It should be noted that the maximum accuracy of the predictions is in reconstruction of the disaggregation. Thus, error values smaller than the reconstruction error values can only be undercut by chance, but not systematically. The developed prediction model is a very simple ANN for a high number of input and target features. Therefore, further optimization in terms of the model of the neural network and perhaps the use of LSTM layers or convolutional layers could enable better forecasts. The ANN is optimized for the used data. Therefore, the results could be worse, when applied to another data set of state changes. The developed forecast does not rely on an exhaustive rollout of measuring frequency devices as in [19] and so is easily transferable also with limited measuring infrastructure. Nevertheless, the transferability must be systematically examined in the future.

It is to be assumed that a certain proportion of state changes of devices during working time is purely coincidental. However, no model can accurately predict randomness. deleted cannot be predicted by any model. In order to assess the chances of success of applying the presented approach to other power data, the randomness of the data must be determined in advance using appropriate methods. For example, the approximated entropy method described in [43] could be used, which has already been applied to contexts such as stock prices, as in [44]. Additionally, instead of a deterministic prediction, one could perform a probabilistic prediction and/or work with confidence intervals for the predicted of power values. This procedure could be helpful in management decision-making.

In this study, we demonstrated the advantages of state changes data for making power predictions. However, the additional informationknowledge from the extraction of device profiles and disaggregation could also be applied to other tasks, such as behavioral analysis, state analyses of buildings, checking the health status of residents or employees, or devising recommendations for intelligent power consumption with respect to the availability of renewable energy. With more variable, market-based electricity tariffs, new business models could even be possible when applying the presented approach to energy management systems. Moreover, the approach could be applied to other data sets other domains, such as households. However, it is currently difficult to obtain three-phase power consumption data that describe active and reactive power, with sufficient temporal granularity to detect events in sn aggregated time series over longer measurement period for the purpose of training the machine learning algorithm used.

6. Conclusions

In this study, we developed an algorithm for extracting device profiles from aggregate power data across six dimensions in a fully unsupervised manner. As the method relies on statistical and unsupervised machine learning methods, it identifies repetitive patterns in aggregate power data. Therefore, the extracted profiles are not necessarily full appliance signatures, but a single operational mode of one device. A direct validation of the device profiles was not possible due to a lack of measured or correct profiles. The transferability of the proposed device profile extraction method is high in theory, as no hyperparameters have to be optimized beforehand, but this must be empirically proven in future studies. The disaggregation uses the extracted device profiles and displays a highly accurate reconstruction. Therefore, the device profiles seemingly represent real appliance signatures sufficiently well. As the final application of the conducted NILM approach, the very-short term power prediction outperformed all of compared predictions. Although many publications developed or carried out various NILM algorithms, the broad application of these methods to other purposes is still missing. In this work, we demonstrated the advantages of the additional information of NILM for very short-term power predictions. Our results and approaches to predictions could be combined with short-term or long-term predictions directly in the power domain. Especially for energy management systems, such combined and high-quality predictions would be of great value for decision-making processes.

Acknowledgments

The authors gratefully acknowledge the financial support of the Federal Ministry for Economic Affairs and Energy of the Federal Republic of Germany for the project *EG2050: EMGIMO: Neue Energieversorgungskonzepte für Mehr-Mieter-Gewerbeimmobilien (03EGB0004G and 03EGB0004A)*. For more details, visit www.emgimo.eu. This study was conducted as part of this project.

References

[1] G. Strbac and A.M. Khambadkone, "Demand side management: Benefits and challenges", *Energy Policy*, 36(12)(2008), pp. 4419-4426

[2] D. Tran and A.M. Khambadkone, "Energy management for lifetime extension of energy storage system in micro-grid applications", *IEEE Transactions on Smart Grid*, 4(3)(2013), pp. 1289-1296

[3] D. Arcos-Aviles, J. Pascual, F. Guinjoan, L. Marroyo, P. Sanchis and M. Marietta, "Low complexity energy management strategy for grid profile smoothing of a residential grid-connected microgrid using generation and demand forecasting", *Applied Energy*, 205(2017), pp. 69-84

[4] C. Wan et.al., "Photovoltaic and solar power forecasting for smart grid energy management", *CSEE Journal of Power and Energy Systems*, 1(4)(2015), pp. 38-46

[5] L. Pedersen, J. Stang and R. Ulseth, "Load prediction method for heat and electricity demand in buildings for the purpose of planning for mixed energy distribution systems", *Energy and Buildings*, 40(7)(2008), pp. 1124-1134

[6] M. Beccali, et.al., "Forecasting daily urban electric load profiles using artificial neural networks", *Energy conversion and management*, 45(18-19)(2004), pp. 2879-2900

[7] H. Li, et.al., "A hybrid annual power load forecasting model based on generalized regression neural network with fruit fly optimization algorithm", *Knowledge-Based Systems*, 37(2013), pp. 378-387

[8] K. Lang, M. Zhang, Y. Yuan, and Y. Xijian, "Short-term load forecasting based on multivariate time series prediction and weighted neural network with random weights and kernels", *Cluster Computing*, vol. 22(2019), pp. 12589-12597

[9] Federal Environment Agency, "Evaluation tables for Energy balance of the Federal Republic of Germany 1990 to 2018", 2019, retrieved from: `https://www.umweltbundesamt.de/daten/energie/stromverbrauch`

[10] G.W. Hart, "Nonintrusive appliance load monitoring", *Proc. of the IEEE*, vol. 80(1992), pp. 1870-1891

[11] J. Kennedy and R. Eberhart, Particle swarm optimization, *Proceedings of ICNN'95 - Int. Conf. on Neural Networks*, pp. 1942-1948 vol. 4, Perth, WA, Australia, 1995

[12] A. Faustine, et. al., "A survey on non-intrusive load monitoring methodies and techniques for energy disaggregation problem." *arXiv preprint arXiv:1703.00785* (2017).

[13] A. Zoha, et. al., "Non-intrusive load monitoring approaches for disaggregated energy sensing: A survey." *Sensors*, 12(12) (2012), pp. 16838-16866.

[14] J.Z. Kolter and M.J. Johnson, "REDD: A public data set for energy disaggregation research", *Proc. SustKDD Workshop on Data Mining Appl. in Sustain.*, 2011

[15] N. Batra, O. Parson, M. Berges, A. Singh, A. Rogers , "A comparison of non-intrusive load monitoring methods for commercial and residential buildings" *arXiv preprint arXiv:1408.6595v1* (2014).

[16] S. Welikala, et. al., "Incorporating appliance usage patterns for non-intrusive load monitoring and load forecasting." *IEEE Transactions on Smart Grid*, 10(1)(2017), pp. 448-461.

[17] M. Wurm and V.C. Coroama, "Grid-level short-term load forecasting based on disaggregated smart meter data" *Computer Science-Research and Development*, 33(1-2)(2018), pp. 265-266.

[18] M.Q. Raza and A. Khosravi, "A review on artificial intelligence based load demand forecasting techniques for smart grid and buildings", *Renewable and Sustainable Energy Reviews*, 50(2015), pp. 1352-1372

[19] A. M. Alonso, F. J. Nogales and C. Ruiz, "A Single Scalable LSTM Model for Short-Term Forecasting of Massive Electricity Time-Series" ,*arXiv preprint arxiv:1910.06640*, 2020

[20] H.S. Hippert, C.E. Pedreira and R.C. Souza, "Neural networks for short-term load forecasting: A review and evaluation", *IEEE Transactions on power systems*, 16(1)(2001), pp. 44-55

[21] K. Brucke, S. Arens, J. Telle, S. Schlüters, B. Hanke, K. Maydell and C. Agert, "Particle Swarm Optimization for Energy Disaggregation in Industrial and Commercial Buildings" ,*arXiv preprint arXiv:2006.12940*, 2020

[22] Janitza electronics GmbH, *Power Quality Analyser UMG 604-PRO - User manual and technical data*, 2017

[23] S.K.K. Ng, J. Liang and J.W.M. Cheng, "Automatic Appliance Load Signature Identification by Statistical Clustering", *8th International Conference on Advances in Power System Control, Operation and Management (APSCOM 2009)*, Hong Kong, pp. 1-6

[24] M. Zeifman, S.R. Shaw and J.L. Kirtley, "Disaggregation of home energy display data using probabilistic approach", *IEEE Transactions on Consumer Electronics*, 58(1)(2012), pp. 23-31

[25] C.M. Bishop, "K-means Clustering", *Pattern Recognition and Machine Learning*, Springer, New York, 2006

[26] T. Caliński and J. Harabasz, "A dendrite method for cluster analysis", *Communications in Statistics - Theory and Methods*, 3(1)(1974), pp. 1-27

[27] R.G. McClarren, "Pearson Correlation", *Uncertainty Quantification and Predictive Computational Science*, Springer, Cham, Switzerland, 2018

[28] C.M. Bishop, "Mixture of Gaussians", *Pattern Recognition and Machine Learning*, Springer, New York, 2006

[29] C.M. Bishop, "EM for Gaussian Mixtures", *Pattern Recognition and Machine Learning*, Springer, New York, 2006

[30] C.M. Bishop, "Model Comparison an BIC", *Pattern Recognition and Machine Learning*, Springer, New York, 2006

[31] S. Amri, W. Barhoumi, E. Zagrouba "Unsupervised background reconstruction based on iterative median blending and spatial segmentation", *Proceedings of IEEE International Conference on Imaging Systems and Techniques, IST 2010*, Thessaloniki, Greece, pp. 411-416

[32] S. Al-Dahidi, O. Ayadi, M. Alrbai and J. Adeeb, "Ensemble approach of optimized artificial neural networks for solar photovoltaic power prediction", *IEEE Access*, vol. 7(2019), pp. 81741-81758

[33] T. Kim and S. Cho, "Predicting residential energy consumption using CNN-LSTM neural networks", *Energy*, vol. 182 (2019), pp. 72-81

[34] A. Torabi, S.A.K. Mousavy, V. Dashti, M. Saeedi and N. Yousefi, "A new prediction model based on cascade NN for wind power prediction", *Computational Economics*, vol. 182(3) (2019), pp. 1219-1243

[35] F. Chollet, "Keras", 2015, retrieved from `https://github.com/fchollet/keras`

[36] T. Steens, J. Telle, B. Hanke, K. Maydell, C. Agert, G. di Modica, B. Engelb and M. Grottke, "A Forecast Based Load Management Approach For Commercial Buildings - Comparing LSTM And Standardized Load Profile Techniques" ,*arXiv preprint arXiv:2007.06832*, 2020

[37] Autonomio Talos [Computer software], 2019, retrieved from `http://github.com/autonomio/talos`

[38] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever and R. Salakhutdinov, "Dropout: A Simple Way to Prevent Neural Networks from Overfitting", *Journal of Machine Learning Research*, 15(2014), pp. 1929-1958

[39] G. Balzer and C. Neumann, "Switching operations [in German]", *Switching and balancing operations in electrical networks [in German]*, Berlin, Germany, Springer, 2018

[40] S.B. Leeb, S.R. Shaw and J.L. Kirtley, "Transient event detection in spectral envelope estimates for nonintrusive load monitoring", *IEEE Transactions on Power Delivery*, 10(3)(1995), pp. 1200-1210

[41] S.B. Leeb, S.R. Shaw and J.L. Kirtley, "Load identification in nonintrusive load monitoring using steady-state and turn-on transient energy algorithms", *The 2010 14th International Conference on Computer Supported Cooperative Work in Design*, , Shanghai, China, pp. 27-32

[42] M. Pal, R. Roy, J. Basu and M.S. Bepari, "Blind source separation: A review and analysis", *2013 International Conference Oriental COCOSDA held jointly with 2013 Conference on Asian Spoken Language Research and Evaluation (O-COCOSDA/CASLRE)*, Gurgaon, 2013, pp. 1-5

[43] S. Pincus, "Approximate entropy as a measure of system complexity", *Proceedings of the National Academy of Sciences*, 8(6)(1991), pp. 2297-2301

[44] A. Delgado-Bonal, "Quantifying the randomness of the stock markets", *Scientific reports*, 9(1)(2019), pp. 1-11